A parallel VNS scheme with ILP neighbourhoods.  
Application to a discrete Unit Commitment Problem

N. Dupin$^{1,2}$

$^1$ University Bordeaux1, 351 cours de la libération, 33400 Talence, France  
$^2$ EDF-R&D, dep. OSIRIS, 1 avenue du Général de Gaulle, 92140 Clamart, France  
nicolas.dupin@math.u-bordeaux1.fr

Integer Linear Programming (ILP) solvers made recent progress with heuristics based on the linear relaxation and variable fixing, see [4] for a survey on these heuristics. On one hand, it allows to cut off earlier branches of the Branch & Bound tree, and on an other hand, very good solutions are found quickly, that allowes to uses an ILP solver in a heuristic mode with a time limit.

Local search heuristics improve a given solution, with modification allowed in a defined neighbourhood. Difficulties come with local extrema, where no local improvement is possible. Variable Neighbourhood Search (VNS) basic idea is to consider different types of neighbourhoods, and to change systematically of neighbourhood within a local search, to have less local extrema.

These paper shows how an ILP solver in a heuristic mode can be very useful and effective in a VNS scheme, considering for neighbourhoods small Integer Linear Programs. These concepts are applied on a discrete Unit Commitment Problem (UCPd), described and modeled in ILP in [1].

1 Parallel VNS scheme with ILP neighbourhoods

The method requires an initial solution. Our VNS scheme improve this solution with a steepest descent procedure, with variable neighbourhoods. The neighbourhoods are defined by fixing some variables in the original ILP to its value in the current best solution. It assures that the considered ILP is feasible, and the current solution is given to the ILP solver as warmstart. The solution provided after a defined resolution time limit (depending on the types of neighbourhood) is thus the next current best current solution. A large number of neighbourhoods types can thus be defined, and by changing systematically the type of neighbourhood, there are very few local optima with this method. The stopping criteria could be a defined time limit, a defined number of iteration without improvement, or till there is no improvement on different types of neighbourhoods.

This heuristic method is generic for an ILP, and easy to implement. Indeed, solution costs are computed directly with the ILP solver, and the current best solution stays always feasible for all the initial constraints, which is convenient for very constrained problems. Furthermore, various types of large neighbourhoods can be defined with this method, and there are fewer local extrema compared to classical metaheuristics. However, these local improvements are more time consuming than other heuristics, it is balanced with the large size of the neighbourhoods, and the progress on the ILP primal heuristics to have quickly good solutions in these large neighbourhoods.

An ILP solver like Cplex is convenient for such use. Fixing variables in the original ILP leads to a small ILP after preprocessing and variable elimination, specially when variable fixing implies other fixing through some coupling constraints. Here, we just need to find quickly good solutions, so the resolution parameters are tuned to use as much as possible the ILP heuristics and the branching strategies to provide good solutions in the short resolution time. A generic resolution mode which emphasizes the primal bound can be generally defined for Cplex. To go further, it is possible to redefine the frequency calls of primal heuristics, or to limit cutting planes passes. Finally, we note that defining the initial solution as warmstart accelerates Cplex resolution.

There are various fixing strategies to define neighbourhoods, following the problem structure. Some strategies can be generic. A first example would be RINS neighbourhoods, similarly to [2]: the fixed variables are the integer variables common in the LP relaxation and in the current solution. An other example would be for multi index variables to restrict the optimization on a subset for one type of index. This can lead to generic randomized neighbourhoods.

This VNS approach is based on the diversity of possible neighbourhoods, so that there are very few local minima. A local optimum for our heuristic is indeed a local optima for all the considered neighbourhoods. To accelerate the convergence of the solution, this approach can be
naturally parallelized, considering parallely different types of neighbourhoods. This is specially useful when very good solutions are found, and few neighbourhoods lead to improvement. In general this approach is well fitted to improve very good solutions, and to leave local optima.

2 ILP neighbourhoods for UCPd

We apply our VNS scheme on the problem UCPd, defined in [1], planning the electricity production at minimum production cost, and fulfilling operating constraints for power plants and the demand in power. The production is discrete, on operating points, so the demands contraints are 0-1 knapsack constraints, the problem has an underlying knapsack structure at every time step. Dynamic constraints model operating constraints, coupling these knapsack problems.

UCPd is modeled as an ILP with binary variables, \( l_{u,t}^{(i)} \), defined for all power plant \( u \), time step \( t \) and operating point \( i \). \( l_{u,t}^{(i)} = 1 \) if the generating power is greater than the power level \( i \), this modeling choice leads to efficient standart branching. The ILP formulation in [1] provide a good quality LP relaxation, and ILP primal heuristics are efficient for this formulation.

Various specific ILP neighbourhoods are defined for UCPd. Through dynamic constraints, variable fixing induces that other variable are fixed, leading to much smaller ILP. RINS generic neighbourhouds can be constructed with the LP relaxation, or after the improvement of cutting planes passes, leading to small neighbourhoods without intuitive.

The multi-index generic fixing has a physical sense. With indexes \( u \), some power plants are selected to be reoptimized, one particular case is to remove the power plants not used in the current solution or the LP relaxation (intuitively with too high production costs). With index \( t \), we can define time windows where the planning can be modified. With index \( i \), the first level, corresponds to fix the set up decisions on the current solution.

Specialized neighbourhoods can be defined. Using the dynamic, of the problem, a variable \( l_{u,t}^{(i)} \) can be fixed if \( l_{u,t}^{(i)} = l_{u,t-1}^{(i)} = l_{u,t+1}^{(i)} \), which corresponds to shift the current planning decisions. The same strategy can be used for index \( i \), it has a sense with the constraints \( l_{u,t}^{(i)} \geq l_{u,t}^{(i+1)} \).

We note that we can combine these fixing strategies, to have a huge diversity of possible neighbourhoods. Furthermore we can add "pseudo cuts" in our neighbourhoods, considering for example that a power plant starting up or shutting down is definitive, to avoid start up costs.

3 Results and perspectives

The implementation used OPL modeling language for solving ILP with Cplex. On real world size instances, our VNS scheme was pretty effective, being able to improve quickly very good solutions founds with the ILP resolution after a very long time.

Our work presents a bridge between metaheuristics and mathematical programming, it offers new perspectives. On one hand, it could be applied in a Branch&Bound method to improve the primal solution found with ILP heuristics, allowing to cut off earlier some branches. On the other hand, it can be used in a classical metaheuristic to leave local optima. Both applications can take advantage of parallelization.

References

5. Cplex, reference manual